Intrinsic Image Decomposition Using Structure-Texture Separation and Surface Normals

Junho Jeon[1], Sunghyun Cho[2,⋆], Xin Tong[3], and Seungyong Lee[1]

[1] POSTECH
[2] Adobe Research
[3] Microsoft Research Asia

Abstract. While intrinsic image decomposition has been studied extensively during the past a few decades, it is still a challenging problem. This is partly because commonly used constraints on shading and reflectance are often too restrictive to capture an important property of natural images, i.e., rich textures. In this paper, we propose a novel image model for handling textures in intrinsic image decomposition, which enables us to produce high quality results even with simple constraints. We also propose a novel constraint based on surface normals obtained from an RGB-D image. Assuming Lambertian surfaces, we formulate the constraint based on a locally linear embedding framework to promote local and global consistency on the shading layer. We demonstrate that combining the novel texture-aware image model and the novel surface normal based constraint can produce superior results to existing approaches.

Keywords: intrinsic image decomposition, structure-texture separation, RGB-D image.

1 Introduction

Intrinsic image decomposition is a problem to decompose an image I into its shading layer S and reflectance layer R based on the following model:

$$I(p) = S(p) \cdot R(p) \tag{1}$$

where p is a pixel position. Shading $S(p)$ at p depicts the amount of light reflected at p, and reflectance $R(p)$ depicts the intrinsic color of the material at p, which is invariant to illumination conditions.

Intrinsic image decomposition has been extensively studied in computer vision and graphics communities because it can benefit many computer graphics and vision applications, such as image relighting [1, 2] and material property editing [3]. Since Land and McCann first introduced Retinex algorithm in 1971 [4], various approaches have been introduced for the last a few decades [5–7]. However, intrinsic image decomposition is still challenging because it is a significantly

⋆ Sunghyun Cho is now with Samsung Electronics.

D. Fleet et al. (Eds.): ECCV 2014, Part VII, LNCS 8695, pp. 218–233, 2014.

ill-posed problem where there are two unknowns $S(p)$ and $R(p)$ for one observed data $I(p)$ at each pixel p.

To overcome the ill-posedness, previous methods use constraints, or priors, on shading and reflectance. Shading has been often assumed to be locally smooth, while reflectance assumed to be piecewise constant [4]. While these assumptions work well on simple cases such as Mondrian-like images consisting of patches with constant reflectance, they fail on most natural images. One important characteristic of natural images is their rich textures. Such textures may be due to the reflectance layer (e.g., a flat surface with dotted patterns), or the shading layer (e.g., a surface with bumps and wrinkles causing a shading pattern). Thus, enforcing simple constraints on either or both shading or reflectance layers with no consideration on textures may cause erroneous results.

In this paper, we propose a novel intrinsic image decomposition model, which explicitly models a separate texture layer T, in addition to the shading layer S and the reflectance layer R. By explicitly modeling textures, S and R in our model depict only textureless base components. As a result, we can avoid ambiguity caused by textures, and use simple constraints on S and R effectively.

To further constrain the problem, we also propose a novel constraint based on surface normal vectors obtained from an RGB-D image. We assume that illumination changes smoothly, and surfaces are Lambertian, i.e., shading of a surface can be determined as a dot product of a surface normal and the light direction. Based on this assumption, our constraint is designed to promote both *local* and *global* consistency of the shading layer based on a locally linear embedding (LLE) framework [8]. For robustness against noise and efficient computation, we sparsely sample points for the surface normal constraint based on local variances of surface normals. This sparse sampling works effectively thanks to our explicit texture modeling. Our shading and reflectance layers do not have any textures, so information from sampled positions can be effectively propagated to their neighbors during the optimization process.

2 Related Work

2.1 Intrinsic Image Decomposition

Intrinsic image decomposition is a long-standing problem in computer vision. The "Retinex" algorithm was first proposed by Land and McCann in 1971 [4]. The algorithm assumes Mondrian-like images consisting of regions of constant reflectances, where large image gradients are typically caused by reflectance changes, and small gradients are caused by illumination. This algorithm was extended to 2D color images by analyzing derivatives of chromaticity [6].

To overcome the fundamental ill-posedness of the problem, several approaches have been introduced utilizing additional information, such as multiple images [2, 9–11], depth maps [12–14], and user interaction [15]. Lee et al. [12] proposed a method, which takes a sequence of RGB-D video frames acquired from a Kinect camera, and proposed constraints on depth information and temporal consistency. In the setting of single input frame, their temporal constraint

cannot be used. Barron and Malik [13] proposed joint estimation of a denoised depth map and spatially-varying illumination. However, due to the lack of proper texture handling, textures which should belong to either shading or reflectance layer may appear in the other layer, as shown in Sec. 6.

Recently, Chen and Koltun [14] showed that high quality decomposition results can be obtained by properly constraining shading components using surface normals without joint estimation of a denoised depth map. Specifically, they find a set of nearest neighbors for each pixel based on their spatial positions and normals, and then constrain the shading component of each pixel to be similar to those of its neighbors. While our surface normal based constraint is similar to theirs, there are three different aspects. First, our constraint is derived from the Lambertian surface assumption, which is more physically meaningful. Second, our constraint uses not only spatially close neighbors but also distant neighbors, so we can obtain more globally consistent results. Third, due to our confidence-based sparse sampling, our method can be more efficient and robust to noise.

Another relevant work to ours is Kwatra et al.'s shadow removal approach for aerial photos [16], which decomposes an aerial image into shadow and texture components. Based on the properties of shadows and textures in aerial images, they define entropy measures for shadows and textures, and minimize them for decomposition. While their work also explicitly considers textures as we do, their work focuses on removal of smooth shadows, such as shadows cast by clouds in aerial images, so it is not suitable for handling complex shadings which are often observed in natural images.

2.2 Texture Separation

Structure-texture separation has also been an important topic and extensively studied. Edge-preserving smoothing has been a popular direction, such as anisotropic filtering [17], total variation [18], bilateral filtering [19], nonlocal means filtering [20], weighted least squares filtering [21], and L^0 smoothing [22]. By applying edge-preserving smoothing to an image, small scale textures can be separated from structure components. However, as these approaches rely on local contrasts to distinguish structures and textures, they may fail to properly capture low contrast structures or high contrast textures.

Other approaches have also been proposed to separate textures regardless of their contrasts. Subr et al. [23] estimate envelopes of local minima and maxima, and average the envelopes to capture oscillatory components. Xu et al. [24] proposed a relative total variation measure, which takes account of inherent variation in a local window. Recently, Karacan et al. [25] proposed a simple yet powerful method, which is based on region covariance matrices [26] and nonlocal means filtering [20]. In our work, we adopt Karacan et al.'s approach to separate textures from shading and reflectance components, as it preserves underlying smooth intensity changes.

3 Image Model and Overall Framework

We define a novel model for intrinsic image decomposition as:

$$I(p) = B(p) \cdot T(p) = S_B(p) \cdot R_B(p) \cdot T(p), \tag{2}$$

where $B(p) = S_B(p) \cdot R_B(p)$ is a base layer, and $S_B(p), R_B(p)$ and $T(p)$ are shading, reflectance and texture components at a pixel p, respectively. Note that S_B and R_B are different from S and R in Eq. (1) as S_B and R_B contain no textures. Based on this model, we can safely assume that R_B is a Mondrian-like image, which is piecewise constant. We also assume that illumination changes smoothly across the entire image, thus S_B is also piecewise smooth with no oscillating variations. Under these assumptions, we will define constraints and energy functions in the following sections, which will be used to decompose an image I into S_B, R_B and T.

The overall process of our method, which consists of two steps, can be summarized as follows. In the first step, we decompose an input RGB image I into a base layer B and a texture layer T. In the second step, the base layer B is further decomposed into a reflectance layer R_B and a shading layer S_B based on the surface normal constraint and other simple constraints. While we use simple constraints similar to previous decomposition methods assuming Mondrian-like images, the constraints can work more effectively as our input for decomposition is B, instead of I, from which textures have been removed. In addition, our global constraint based on surface normals promotes overall consistency of the shading layer, which is hard to achieve by previous methods. Experimental results and comparisons in Sec. 6 demonstrate the effectiveness of our method.

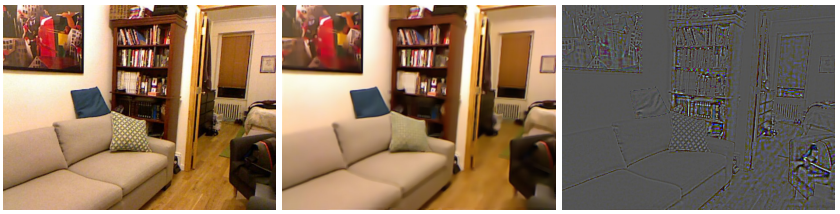
4 Decomposition of B and T

In this step, we decompose an input image I into a base layer B and a texture layer T. Texture decomposition has been studied extensively for long time, and several state-of-the-art approaches are available. Among them, we adopt the region covariance based method of Karacan et al. [25], which performs nonlocal means filtering with patch similarity defined using a region covariance descriptor [26]. This method is well-suited for our purpose as it preserves the smoothly varying shading information in the filtering process. We also tested other methods, such as [22, 24], which are based on total variation, but we found that they tend to suppress all small image gradients, including those from shading. Fig. 1 shows that the region covariance based method successfully removes textures on the cushion and the floor while preserving shading on the sofa.

5 Decomposition of S and R

After obtaining B, we decompose it into S_B and R_B by minimizing the following energy function:

$$f(S_B, R_B) = f^N(S_B) + \lambda^P f^P(S_B) + \lambda^R f^R(R_B) \tag{3}$$



(a) Input image (b) Base layer B (c) Texture layer T

Fig. 1. Decomposition of B and T

subject to $B(p) = S_B(p) \cdot R_B(p)$. In Eq. (3), $f^N(S_B)$ is a surface normal based constraint on S_B, $f^P(S_B)$ is a local propagation constraint on S_B, and $f^R(R_B)$ is a piecewise constant constraint on R_B. In the following, we will describe each constraint in more detail.

5.1 Surface Normal Based Shading Constraint $f^N(S_B)$

LLE-Based Local Consistency. To derive the surface normal based shading constraint, we first assume that surfaces are Lambertian. On a Lambertian surface, shading S and a surface normal N at p have the following relation:

$$S(p) = i_L \cdot \langle L(p), N(p) \rangle \,, \tag{4}$$

where i_L is an unknown light intensity, $L(p)$ is the lighting direction vector at p, and $\langle L(p), N(p) \rangle$ is the inner product of $L(p)$ and $N(p)$.

As we assume that illumination changes smoothly, we can also assume that i_L and L are constant in a local window. We can express a surface normal at p as a linear combination of normals at its neighboring pixels $q \in \mathcal{N}_l(p)$, i.e. $N(p) = \sum_{q \in \mathcal{N}_l(p)} w^N_{pq} N(q)$ where w^N_{pq} is a linear combination weight. Then, $S(p)$ can also be expressed as the same linear combination of the neighbors $S(q)$:

$$S(p) = i_L \cdot \left\langle L, \sum_{q \in \mathcal{N}_l(p)} w^N_{pq} N(q) \right\rangle = \sum_{q \in \mathcal{N}_l(p)} w^N_{pq} \left(i_L \cdot \langle L, N(q) \rangle \right) = \sum_{q \in \mathcal{N}_l(p)} w^N_{pq} S(q). \tag{5}$$

Based on this relation, we can define a local consistency constraint $f^N_l(S_B)$ as:

$$f^N_l(S_B) = \sum_{p \in \mathcal{P}_N} \left(S_B(p) - \sum_{q \in \mathcal{N}_l(p)} w^N_{pq} S_B(q) \right)^2 , \tag{6}$$

where $\mathcal{P}_N$ is a set of pixels. Note that we could derive this constraint without having to know the value of the light intensity i_L.

Interestingly, Eq. (5) can be interpreted as a LLE representation [8]. LLE is a data representation, which projects a data point from a high dimensional space onto a low dimensional space by representing it as a linear combination of its

neighbors in the feature space. Adopting the LLE approach, we can calculate the linear combination weights $\{w_{pq}^N\}$ by solving:

$$\underset{\{w_{pq}^N\}}{\operatorname{argmin}} \sum_{p \in \mathcal{P}_N} \|N(p) - \sum_{q \in \mathcal{N}_l(p)} w_{pq}^N N(q)\|^2, \tag{7}$$

subject to $\sum_{q \in \mathcal{N}_l(p)} w_{pq}^N = 1$.

To find $\mathcal{N}_l(p)$, we build a 6D vector for each pixel as $[x(p), y(p), z(p), n_x(p), n_y(p), n_z(p)]^T$, where $[x(p), y(p), z(p)]^T$ is the 3D spatial location obtained from the input depth image, and $[n_x(p), n_y(p), n_z(p)]^T$ is the surface normal at p. Then, we find the k-nearest neighbors using the Euclidean distance between the feature vectors at p and other pixels.

Global Consistency. While a locally consistent shading result can be obtained with $f_l^N(S_B)$, the result may be still globally inconsistent. Imagine that we have two flat regions, which are close to each other, but their depths are slightly different. Then, for each pixel in one region, all of its nearest neighbors will be found in the same region, and the two regions may end up with completely different shading values. This phenomenon can be found in Chen and Koltun's results in Fig. 6, as their method promotes only local consistency. In their shading result on the first row, even though the cushion on the sofa should have similar shading to the sofa and the wall, they have totally different shading values.

In order to avoid such global inconsistency, we define another constraint $f_g^N(S_B)$, which promotes global consistency:

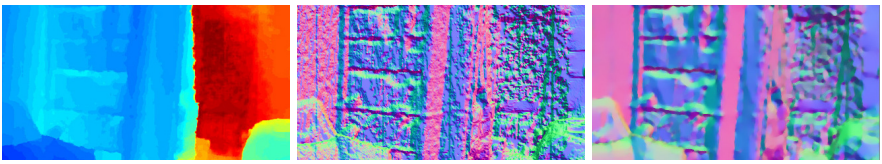
$$f_g^N(S_B) = \sum_{p \in \mathcal{P}_N} \left(S_B(p) - \sum_{q \in \mathcal{N}_g(p)} w_{pq}^N S_B(q) \right)^2, \tag{8}$$

where $\mathcal{N}_g(p)$ is a set of global k-nearest neighbors for each pixel p. To find $\mathcal{N}_g(p)$, we simply measure the Euclidean distance between the surface normals at p and other pixels without considering their spatial locations, so that the resulting $\mathcal{N}_g(p)$ can include spatially distant pixels. With the two constraints, we define the constraint $f^N(S_B)$ as:

$$f^N(S_B) = f_l^N(S_B) + \lambda_g^N f_g^N(S_B), \tag{9}$$

where λ_g^N is the relative weight for f_g^N. We set $k = 20$ for both local and global consistency constraints.

Sub-sampling for Efficiency and Noise Handling. It can be time-consuming to find k-nearest neighbors and apply $f^N(S_B)$ for every pixel in an image. Moreover, depth images from commercial RGB-D cameras are often severely noisy as shown in Fig. 2a. We may apply a recent depth map denoising method, but there can still remain significant errors around depth discontinuities causing a noisy normal map (Fig. 2c).



(a) Raw depth image (b) Raw normal map (c) Denoised normal map

Fig. 2. Depth images from commercial RGB-D cameras often suffer from severe noise, which is difficult to remove using a denoising method

To improve the efficiency and avoid noisy normals, we propose a sub-sampling based strategy for building $\mathcal{P}_N$. Specifically, we divide an image into a uniform grid. In each grid cell, we measure the variance of the surface normals in a local window centered at each pixel. Then, we choose a pixel with the smallest variance. This is because complex scene geometry is more likely to cause severe depth noise, so we would better choose points in a smooth region with low variance. We also find the nearest neighbors for $\mathcal{N}_l(p)$ and $\mathcal{N}_g(p)$ from $\mathcal{P}_N$ to avoid noisy normals and accelerate the nearest neighbor search. While we use the constraint $f^N(S_B)$ only for sub-sampled pixel positions, information on the sampled positions can be propagated to neighboring pixels during the optimization due to the constraint $f^P(S_B)$, which is described next.

5.2 Local Propagation Constraint $f^P(S_B)$

Since we use subsampled pixel positions for the constraint $f^N(S_B)$, other pixels do not have any shading constraint. To properly constrain such pixels, we propagate the effects of $f^N(S_B)$ to neighboring pixels using two local smoothness constraints on shading. Specifically, we define $f^P(S_B)$ as:

$$f^P(S_B) = f^P_{lap}(S_B) + \lambda^P_N f^P_N(S_B), \tag{10}$$

where $f^P_{lap}(S_B)$ is based on the structure of the base layer B, and $f^P_N(S_B)$ is based on surface normals.

Since all the textures are already separated out to T and we assume that R_B is piecewise constant, we can safely assume that small image derivatives in B are from the shading layer S_B. Then, S_B can be approximated in a small local window as:

$$S_B(p) \approx aB(p) + b, \tag{11}$$

where $a = \frac{1}{B_f - B_b}$ and $b = \frac{-B_b}{B_f - B_b}$, and B_f and B_b are two primary colors in the window. This approximation inspires us to use the matting Laplacian [27] to propagate information from the sub-sampled pixels to their neighbors in a structure-aware manner. Specifically, we define the first constraint for propagation using the matting Laplacian as follows:

$$f^P_{lap}(S_B) = \sum_k \sum_{(i,j)\in\omega_k} w^{lap}_{ij}(S_B(i) - S_B(j))^2, \tag{12}$$

where ω_k is the k-th local window. w_{ij}^{lap} is the (i,j)-th matting Laplacian element, which is computed as:

$$w_{ij}^{lap} = \sum_{k|(i,j)\in\omega_k} \left\{ \delta_{ij} - \frac{1}{|\omega_k|} \left(1 + \left(B(i) - \mu_k^B \right) \left(\Sigma_k^B + \frac{\epsilon}{|\omega_k|} I_3 \right)^{-1} \left(B(j) - \mu_k^B \right) \right) \right\}, \tag{13}$$

where ϵ is a regularizing parameter, and $|\omega_k|$ is the number of pixels in the window ω_k. δ_{ij} is Kronecker delta. μ_k^B and Σ_k^B are the mean vector and covariance matrix of B in ω_k, respectively. I_3 is a 3×3-identity matrix.

This constraint is based on the key advantage of removing textures from the input image. With the original image I, because of textures, information at sample points cannot be propagated properly while being blocked by edges introduced by textures. In contrast, by removing textures from the image, we can effectively propagate shading information using the structure of the base image B, obtaining higher quality shading results.

The second local smoothness constraint $f_N^P(S_B)$ is based on surface normals. Even if surface normals are noisy, they still provide meaningful geometry information for smooth surfaces. Thus, we formulate a constraint to promote local smoothness based on the differences between adjacent surface normals as follows:

$$f_N^P(S_B) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{N}(p)} w_{pq}^N \left(S_B(p) - S_B(q) \right)^2, \tag{14}$$

where $\mathcal{P}$ is a set of all pixels in the image and $\mathcal{N}(p)$ is the set of 8-neighbors of p. w_{pq}^N is a continuous weight, which is defined using the angular distance between normal vectors at p and q:

$$w_{pq}^N = \exp\left(-\frac{1 - \langle N(p), N(q) \rangle^2}{\sigma_n^2} \right), \tag{15}$$

where we set $\sigma_n = 0.5$. w_{pq}^N becomes close to 1 if the normals $N(p)$ and $N(q)$ are similar to each other, and becomes small if they are different.

Fig. 3 shows the effect of the constraint $f^P(S_B)$. The local propagation constraint enables shading information obtained from the LLE-based constraints to be propagated to other pixels, which results in clear shading near edges.

5.3 Reflectance Constraint $f^R(R_B)$

The constraint $f^R(R_B)$ is based on a simple assumption, which is used by many Retinex-based approaches [6, 12, 28, 29]: if two neighboring pixels have the same chromaticity, their reflectance should be the same as well. Based on this assumption, previous methods often use a constraint defined as:

$$f^R(R_B) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{N}(p)} w_{pq}^R \left(R_B(p) - R_B(q) \right)^2. \tag{16}$$

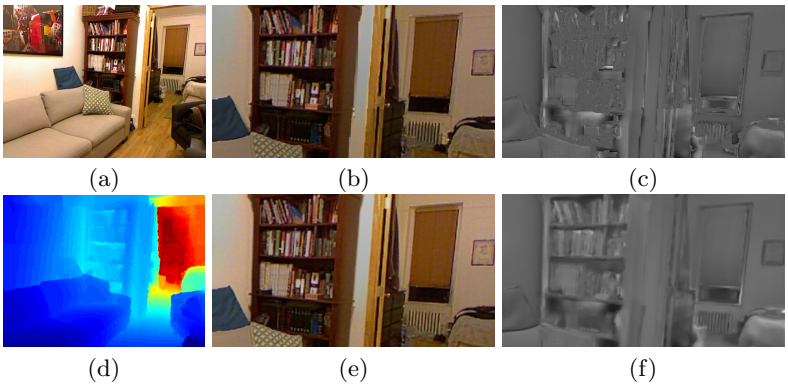


Fig. 3. Effect of the local propagation constraint $f^P(S_B)$. (a,d) Input RGB and depth images. (b-c) Reflectance and shading results without $f^P(S_B)$. (e-f) Reflectance and shading results with $f^P(S_B)$. Without the constraint, shading of pixels which have not been sampled for normal based constraint $f^N(S_B)$ are solely determined by the reflectance constraint through the optimization process. As a result, (c) shows artifacts on the bookshelves due to the inaccurate reflectance values.

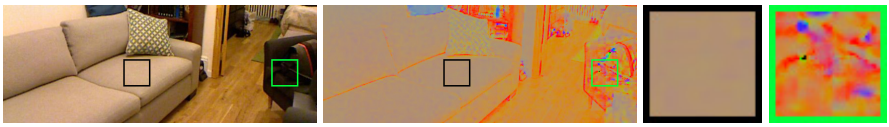
The weighting function w^R_{pq} is a positive constant if the difference between the chromaticity at p and q is smaller than a certain threshold, and zero otherwise. For this constraint, it is critical to use a good threshold, but finding a good threshold is non-trivial and can be time consuming [29].

Instead of using a threshold, we use a continuous weight $w^{R'}_{pq}$, which involves chromaticity difference between two pixels p and q in the form of angular distance between two directional vectors:

$$w^{R'}_{pq} = \exp\left(-\frac{1-\langle C(p), C(q)\rangle^2}{\sigma_c^2}\right)\left\{1+\exp\left(-\frac{B(p)^2+B(q)^2}{\sigma_i^2}\right)\right\}, \tag{17}$$

where chromaticity $C(p) = B(p)/|B(p)|$ is a normalized 3D vector consisting of RGB color channels. We set $\sigma_c^2 = 0.0001, \sigma_i^2 = 0.8$. The first exponential term measures similarity between $C(p)$ and $C(q)$ using their angular distance. The term becomes 1 if $C(p) = C(q)$ and becomes close to 0 if $C(p)$ and $C(q)$ are different from each other, so that consistency between neighboring pixels can be promoted only when $C(p)$ and $C(q)$ are close to each other. The second exponential term is a darkness weight. Due to the nature of imaging sensors, dark pixels suffer from noise more than bright pixels, causing severe chromaticity noise in dark regions such as shadows (Fig. 4). Thus, the second term gives larger weights to dark pixels to overcome such chromaticity noise. Using the weight $w^{R'}_{pq}$, our constraint $f^R(R_B)$ is defined as:

$$f^R(R_B) = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{N}(p)} w^{R'}_{pq}\left(R_B(p) - R_B(q)\right)^2. \tag{18}$$



(a) (b) (c)

Fig. 4. Comparison of chromaticity noise between a bright region (black box) and a dark region (green box). (a) Input image. (b) Chromaticity of the input image. (c) Magnified views of two regions.

5.4 Optimization

To simplify the optimization, we take the logarithm to our model as done in [29, 12, 14]. Then, we get $\log B(p) = s_B(p) + r_B(p)$ where $s_B(p) = \log S_B(p)$ and $r_B(p) = \log R_B(p)$. We empirically found that proposed constraints could also represent the similarities between pixels in the logarithmic domain, even for the LLE weights. Thus, we approximate the original energy function (Eq. (3)) as:

$$f(s_B) = f^N(s_B) + \lambda^P f^P(s_B) + \lambda^R f^R(\log B(p) - s_B(p)). \tag{19}$$

As all the constraints f^N, f^P, and f^R are quadratic functions, Eq. (19) is a quadratic function of s_B. We minimize Eq. (19) using a conjugate gradient method. We used $\lambda^P = \lambda^R = 4, \lambda_g^N = 1$, and $\lambda_N^P = 0.00625$.

6 Results

For evaluation, we implemented our method using Matlab, and used the structure-texture separation code of the authors [25] for decomposition of B and T. For a 624×468 image, decomposition of B and T takes about 210 seconds, and decomposition of S and R takes about 150 seconds on a PC with Intel Core i7 2.67GHz CPU, 12GB RAM, and Microsoft Windows 7 64bit OS.

For evaluation, we selected 14 images from the NYU dataset [30], which provides 1,400 RGB-D images. When selecting images, we avoided images which do not fit our Lambertian surface assumption, e.g., images with glossy surfaces such as a mirror or a glass. Fig. 5 shows some of the selected images. For other images and their corresponding results, we refer the readers to our supplementary material. It is worth mentioning that, while Chen and Koltun [14] used the synthetic dataset of [31] to quantitatively evaluate their approach, in our evaluation, we did not include such quantitative evaluation. This is because the synthetic dataset of [31] was not generated for the purpose of intrinsic image decomposition benchmark, so its ground truth reflectance and shading images are not physically correct in many cases, such as shading of messy hairs and global illumination.

In our image model $I(p) = S_B(p) \cdot R_B(p) \cdot T(p)$, texture T can contain not only reflectance textures, but also shading textures caused by subtle and complex geometry changes, which are often not captured in a noisy depth map. In this

Fig. 5. Test images from the NYU dataset [30]

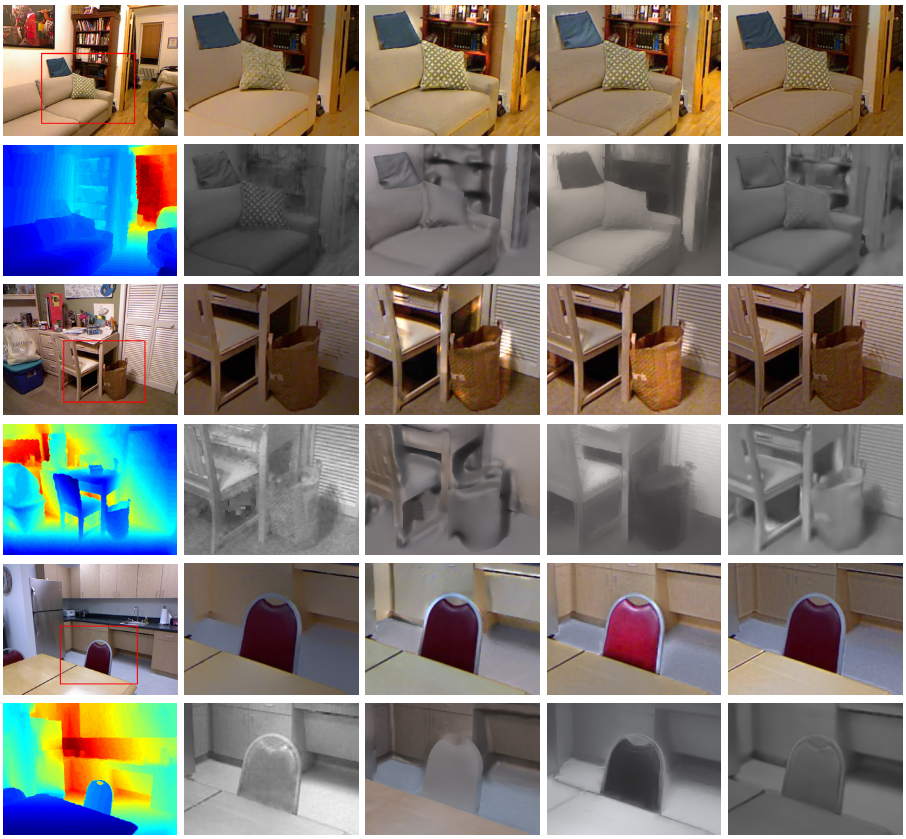
paper, we do not further decompose T into reflectance and shading textures. Instead, for visualization and comparison, we consider T as a part of the reflectance layer R, i.e., $R(p) = R_B(p) \cdot T(p)$ and $S(p) = S_B(p)$. That is, every reflectance result in this paper is the product of the base reflectance layer R_B and the texture layer T.

We evaluated our method using qualitative comparisons with other methods. Fig. 6 shows results of three other methods and ours. The conventional color Retinex algorithm [32] takes a single RGB image as an input. This method highly depends on its reflectance smoothness constraint, which can be easily discouraged by rich textures. Thus, its shading results contain a significant amount of textures, which should be in the reflectance layers (e.g., patterned cushion in the sofa scene).

Barron and Malik [13] jointly estimate a refined depth map and spatially varying illumination, and obtain a shading image from that information. Thus, their shading results in Fig. 6 do not contain any textures on them. However, their shading results are over-smoothed on object boundaries because of their incorrect depth refinement (e.g., the chair and bucket in the desk scene).

The results of Chen and Koltun [14] show more reliable shading results than [32, 13], even though they do not use any explicit modeling for textures. This is because of their effective shading constraints based on depth cues. However, as mentioned in Sec. 2, due to the lack of global consistency constraints, their shading results often suffer from the global inconsistency problem (e.g., the chair in the kitchen scene). On the contrary, our method produces well-decomposed textures (e.g., cushion in the sofa scene) and globally consistent shading (e.g., the chair and the bucket in the desk scene) compared to the other three methods.

To show the effect of our texture filtering step, we also tested our algorithm without texture filtering (Fig. 7). Thanks to our non-local shading constraints, the shading results are still globally consistent (Fig. 7b). However, the input image without texture filtering breaks the Mondrian-like image assumption, so lots of reflectance textures remain in the shading result. This experiment shows that our method fully exploits properties of the texture filtered base image, such as piecewise-constant reflectance and texture-free structure information.

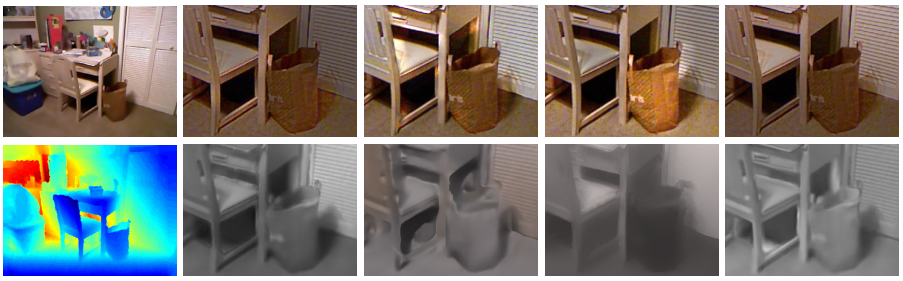


(a) (b) (c) (d) (e)

Fig. 6. Decomposition result comparison with other methods. (a) Input. (b) Retinex [32]. (c) Barron and Malik [13]. (d) Chen and Koltun [14]. (e) Ours.

(a) (b) (c)

Fig. 7. Decomposition results without and with the texture filtering step. (a) Input image. (b) Reflectance and shading results without texture filtering step. (c) Reflectance and shading results with texture filtering step.



(a) (b) (c) (d) (e)

Fig. 8. Decomposition results of other methods using a texture-filtered input. (a) Input base image. (b) Retinex [32]. (c) Barron and Malik [13]. (d) Chen and Koltun [14]. (e) Ours.

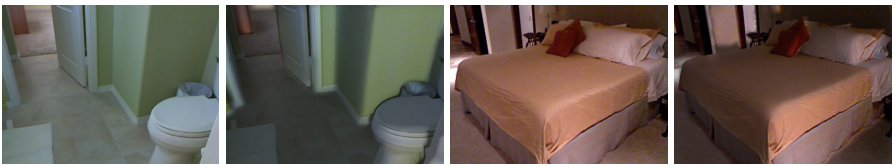
We also conducted another experiment to clarify the effectiveness of our decomposition step. This time, we fed texture-filtered images to previous methods as their inputs. Fig. 8 shows texture filtering provides some improvements to other methods too, but the improvements are not as big as ours. Retinex [32] benefited from texture filtering, but the method has no shading constraints and the result still shows globally inconsistent shading (the bucket and the closet). Big improvements did not happen with recent approaches [13, 14] either. [13] strongly uses its smooth shape prior, which causes over-smoothed shapes and shading in regions with complex geometry. In the result of [14], globally inconsistent shading still remains due to the lack of global consistency constraints.

7 Applications

One straightforward application of intrinsic image decomposition is material property editing such as texture composition. Composing textures into an image naturally is tricky, because it requires careful consideration of spatially varying illumination. If illumination changes such as shadows are not properly handled, composition results may look too flat and artificial (Fig. 9b). Instead, we can first decompose an image into shading and reflectance layers, and compose new textures into the reflectance layer. Then, by recombining the shading and reflectance layers, we can obtain a more naturally-looking result (Fig. 9c).

(a) Original image (b) Naive copy & paste (c) Our method

Fig. 9. Texture composition



(a) (b) (c) (d)

Fig. 10. Image relighting. (a, c) Original images. (b, d) Relighted images.

Another application is image relighting (Fig. 10). Given an RGB-D input image, we can generate a new shading layer using the geometry information obtained from the depth information. Then, by combining the new shading layer with the reflectance layer of the input image, we can produce a relighted image.

8 Conclusions

Although intrinsic image decomposition has been extensively studied in computer vision and graphics, the progress has been limited by the nature of natural images, especially rich textures. In this work, we proposed a novel image model, which explicitly models textures for intrinsic image decomposition. With explicit texture modeling, we can avoid confusion on the smoothness property caused by textures and can use simple constraints on shading and reflectance components. To further constrain the decomposition problem, we additionally proposed a novel constraint based on surface normals obtained from an RGB-D image. Assuming Lambertian surfaces, we formulated our surface normal based constraints using a LLE framework [8] in order to promote both local and global consistency of shading components.

In our experiments, we assumed textures to be a part of reflectance for the purpose of comparison with other methods. However, textures may be caused by either or both of reflectance and shading, as we mentioned in Introduction. As future work, we plan to further decompose textures into reflectance and shading texture layers using additional information such as surface geometry.

Acknowledgements. We would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by IT/SW Creative Research Program of NIPA (2013-H0503-13-1013).

References

1. Yu, Y., Malik, J.: Recovering photometric properties of architectural scenes from photographs. In: Proc. of SIGGRAPH, pp. 207–217. ACM (1998)
2. Laffont, P.Y., Bousseau, A., Paris, S., Durand, F., Drettakis, G., et al.: Coherent intrinsic images from photo collections. ACM Transactions on Graphics 31(6) (2012)

3. Khan, E.A., Reinhard, E., Fleming, R.W., Bülthoff, H.H.: Image-based material editing. ACM Transactions on Graphics 25(3), 654–663 (2006)
4. Land, E.H., McCann, J.J.: Lightness and retinex theory. Journal of the Optical Society of America 61(1) (1971)
5. Barrow, H.G., Tenenbaum, J.M.: Recovering intrinsic scene characteristics from images. Computer Vision Systems (1978)
6. Funt, B.V., Drew, M.S., Brockington, M.: Recovering shading from color images. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 124–132. Springer, Heidelberg (1992)
7. Kimmel, R., Elad, M., Shaked, D., Keshet, R., Sobel, I.: A variational framework for retinex. International Journal of Computer Vision 52, 7–23 (2003)
8. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
9. Weiss, Y.: Deriving intrinsic images from image sequences. In: Proc. of ICCV (2001)
10. Matsushita, Y., Lin, S., Kang, S.B., Shum, H.-Y.: Estimating intrinsic images from image sequences with biased illumination. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part II. LNCS, vol. 3022, pp. 274–286. Springer, Heidelberg (2004)
11. Laffont, P.Y., Bousseau, A., Drettakis, G.: Rich intrinsic image decomposition of outdoor scenes from multiple views. IEEE Transactions on Visualization and Computer Graphics 19(2) (2013)
12. Lee, K.J., Zhao, Q., Tong, X., Gong, M., Izadi, S., Lee, S.U., Tan, P., Lin, S.: Estimation of intrinsic image sequences from image+depth video. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 327–340. Springer, Heidelberg (2012)
13. Barron, J.T., Malik, J.: Intrinsic scene properties from a single RGB-D image. In: Proc. of CVPR (2013)
14. Chen, Q., Koltun, V.: A simple model for intrinsic image decomposition with depth cues. In: Proc. of ICCV (2013)
15. Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. ACM Transactions on Graphics 28(5) (2009)
16. Kwatra, V., Han, M., Dai, S.: Shadow removal for aerial imagery by information theoretic intrinsic image analysis. In: International Conference on Computational Photography (2012)
17. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
18. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60(1-4), 259–268 (1992)
19. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. of ICCV (1998)
20. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. of CVPR (2005)
21. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Transactions on Graphics 27(3), 67:1–67:10 (2008)
22. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L0 gradient minimization. ACM Transactions on Graphics 30(6), 174:1–174:12 (2011)
23. Subr, K., Soler, C., Durand, F.: Edge-preserving multiscale image decomposition based on local extrema. ACM Transactions on Graphics 28(5), 147:1–147:9 (2009)
24. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Transactions on Graphics 31(6), 139:1–139:10 (2012)

25. Karacan, L., Erdem, E., Erdem, A.: Structure-preserving image smoothing via region covariances. ACM Transactions on Graphics 32(6), 176:1–176:11 (2013)
26. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
27. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(2), 228–242 (2008)
28. Shen, L., Yeo, C.: Intrinsic images decomposition using a local and global sparse representation of reflectance. In: Proc. of CVPR (2011)
29. Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., Lin, S.: A closed-form solution to retinex with nonlocal texture constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7) (2012)
30. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
31. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 611–625. Springer, Heidelberg (2012)
32. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: Proc. of ICCV (2009)